

Algoritmo de ordenamiento Quicksort

El algoritmo de ordenamiento Quicksort(Ordenación Rápida) es más rápido y fácil de ordenar dos listas pequeñas que una lista grande. Se denomina método de ordenación rápida porque, en general, puede ordenar una lista de datos mucho más rápidamente que cualquiera de los otros métodos de ordenación .

El método se basa en la estrategia típica de "divide y vencerás" (divide and conquer). La lista a clasificar almacenada en un vector o array se divide (parte) en dos sublistas: una con todos los valores menores o iguales a un cierto valor específico y otra con todos los valores mayores que ese valor. El valor elegido puede ser cualquier valor arbitrario del vector. En ordenación rápida se llama a este valor pivote. El primer paso es dividir la lista original en dos sublistas o subvectores y un valor de separación. Así, el vector $V[]$ se divide en tres partes:

- • Subvector V_{Izq} , que contiene los valores inferiores o iguales.
- • El elemento de separación.
- • Subvector V_{Dch} , que contiene los valores superiores o iguales.

Los subvectores V_{Izq} y V_{Dch} no están ordenados, excepto en el caso de reducirse a un elemento.

Consideremos la lista de valores.

Vector $V[18 \ 11 \ 27 \ 13 \ 9 \ 4 \ 16]$

Se elige un pivote, 13. Se recorre el Array o Vector desde el extremo izquierdo y se busca un elemento mayor que 13 (se encuentra el 18). A continuación, se busca desde el extremo derecho un valor menor que 13 (se encuentra el 4).

$V[18 \ 11 \ 27 \ <13> \ 9 \ 4 \ 16]$

Se intercambian estos dos valores y se produce la lista

$V[4 \ 11 \ 27 \ <13> \ 9 \ 18 \ 16]$

Se sigue recorriendo el vector por la izquierda y se localiza el 27, y a continuación otro valor bajo se encuentra a la derecha (el 9). Intercambiar estos dos valores y se obtiene

$V[4 \ 11 \ 9 \ <13> \ 27 \ 18 \ 16]$

Al intentar este proceso una vez más, se encuentra que las exploraciones de los dos extremos vienen juntos sin encontrar ningún futuro valor que esté "fuera de lugar". En este punto se conoce que todos los valores a la derecha son mayores que todos los valores a la izquierda del pivote. Se ha realizado una partición en la lista original, que se ha quedado dividida en dos listas más pequeñas:

Vizq VDch
V[4 11 9 <13> 27 18 16] //<13> centro o pivote

Ninguno de los subvectores está ordenado; sin embargo, basados en los resultados de esta primera partición, se pueden ordenar ahora las dos particiones independientemente. Esto es, si ordenamos el subvector Vizq[4 11 9] en su posición,

Vdch[27 18 16] en su posición,
Dará Lugar al ordenamiento:

V[4 9 11 13 16 18 27]

El procedimiento de ordenación supone, en primer lugar, una partición del del Vector.